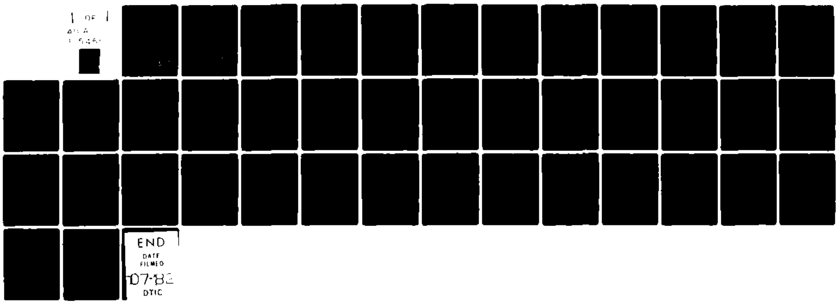
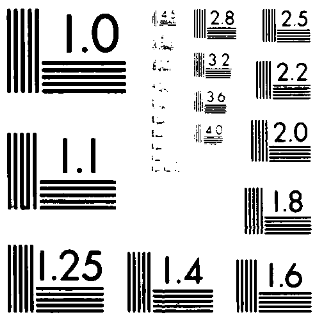
AD-A115 451 NORTH CAROLINA UNIV AT CHAPEL HILL CURRICULUM IN OPER--ETC F/G 12/2
SIMULATING A MARKOV CHAIN WITH A SUPEREFFICIENT SAMPLING METHOD--ETC(U)
APR 82 G S FISHMAN N00014-26-C-0302
UNCLASSIFIED UNC/ORSA/TR-82/3 NL

1 OF 1
AD A
115451



END
DATE
FILMED
07-82
DTIC

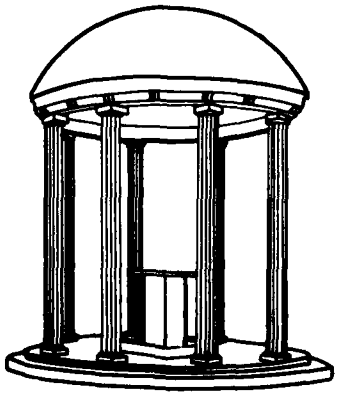

1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

(12)

AD A115431

OPERATIONS RESEARCH AND SYSTEMS ANALYSIS

UNIVERSITY OF NORTH CAROLINA
AT CHAPEL HILL



82 06 10 032

12

SIMULATING A MARKOV CHAIN WITH A SUPEREFFICIENT SAMPLING METHOD

George S. Fishman

Technical Report No. UNC/ORSA/TR-82/3
April 1982

Curriculum in Operations Research
and Systems Analysis
University of North Carolina at Chapel Hill

This research was supported by the Office of Naval Research under contract N00014-26-C-0302. Reproduction in whole or part is permitted for any purpose of the United States government.

Abstract

This paper describes an algorithm and a FORTRAN subprogram, CHAIN, for simulating the behavior of an $(n+1)$ state Markov chain using a variance reducing technique called *rotation sampling*. The simulation of k *microreplications* is carried out in parallel at a mean cost $\lesssim O(\ln k)$ and with variances of sample quantities of interest $\lesssim O((\ln k)^2/k^2)$. The program allows for independent *macroreplications*, each of k microreplications, in order to facilitate estimation of the variances of sample quantities of interest. The paper describes theoretical results that underlie the algorithm and program in Section 1 and presents applications of interest for first passage time and steady-state distributions in Section 2. Section 3 describes the algorithm and CHAIN and an example in Section 4 illustrates how CHAIN works in practice. Section 5 describes the options available for restarting the simulation.

[illegible]

Introduction

A recent paper (Fishman 1981) describes how one can use rotation sampling, a special case of the antithetic variate method, to induce substantial variance reduction in the simulation of a finite state Markov chain. Since many discrete event simulations have an underlying Markov structure or one close to being Markov, this variance reducing proposal has clear appeal. Moreover, for large and possible ill-conditioned transition matrices, one may prefer the Monte Carlo or simulation method with appropriate variance reducing plans to numerical analysis when solving for steady-state and first passage time distributions. In fact, it may be the only feasible method for some problems.

This earlier work derives its cost-saving potential from viewing the simulation of k tours in *series* of a finite $(n+1)$ state positive recurrent aperiodic Markov chain as equivalent to the simulation of k replications of the Markov chain in *parallel*. Although the marginal distributions that arise with the two alternative formulations are necessarily the same for corresponding variables, the parallel formulation allows one to induce joint distributions across replications that lead to a significant cost saving. The induced joint distributions follow from the use of rotation sampling, as described in detail in Fishman and Huang (1980). The cost saving arises in two ways. Firstly, for fixed n, run time in the correlated case is $O(\ln k)$ in contrast to $O(k)$ for the serial simulation. Secondly, for fixed n the variance of an estimator of interest has an upper bound $O((\ln k/k)^2)$ for the correlated case compared to $O(1/k)$ for the serial case.

The present paper describes an algorithm for implementing the essential steps for k parallel simulations with correlation along individual sample paths induced by rotation sampling. The paper also describes a FORTRAN program, called CHAIN, that one can use to perform the simulation. In particular, CHAIN runs I independent *macroreplications* each of K correlated parallel *microreplications* and computes point estimates of interest and sample variances of these point estimates.

Section 1 introduces the Markov chain rotation and describes the results on variance reduction derived in Fishman (1981a) for finite state chains. For completeness it also presents results in Fishman (1981b) for infinite state chains. Section 2 describes several potential uses of CHAIN. These include first passage time distributions and steady-state probabilities for semi-Markov processes. Section 3 presents Procedure MC which contains the essential steps in carrying out parallel simulation based on rotation sampling. It also describes the FORTRAN CHAIN subprogram in detail. Section 4 describes an example of how CHAIN can be used in practice. The example is of the discrete time Markov chain that corresponds to the M/M/1 queueing problem with finite capacity n .

1. Definitions and Previous Results

Consider a positive recurrent aperiodic Markov chain with state space $S = (0,1,2,\ldots,n)$ and transition probabilities $\{p_{ij};\ i,j=0,1,\ldots,n\}$

where there exists a positive integer $\delta \le (n-1)/2$ such that

$$p_{ij} = 0 \qquad \text{for } |\, i-j \,| > \delta$$

and

$$\sum_{j=\max(0,i-\delta)}^{i+\delta} p_{ij} = 1 \quad . \tag{1}$$

It is convenient to describe an alternative, but equivalent, representation to (1) whose value is apparent when actually generating sample paths by simulation on a computer. Let s_j denote the total number of states that have positive transition probabilities from state j and let $\{m_{jr};\ r = 1,\ldots,s_j\}$ denote the ordered sequence $(m_{jr} < m_{j,r+1};\ r = 1,\ldots,s_j - 1)$ of the s_j states to which entry can occur from state j . Then one has the representation

$$p_{jm_{jr}} > 0 \qquad r = 1,\ldots,s_j$$

$$\sum_{r=1}^{s_j} p_{jm_{jr}} = 1$$

$$\delta \ge \max\,(|m_{j1} - j\,|, |\, m_{js_j} - j|) \qquad j = 0,1,\ldots \quad .$$

Suppose one wants to use simulation to study the behavior of the chain during a time period that begins with exit from state a and ends with entry to state $b \in S$. Consider k replications of the simulation experiment run in parallel. Let $K'_{ij\ell}$ denote the number of replications that move from i to j on transition ℓ and let $K'_{j\ell}$ be the number of replications in state j at the end of transition ℓ.† Let $U, U_1, \ldots, U_{K'_{j\ell}}$ be from $U(0,1)$ where $K'_{j\ell} > 0$ is given. Then for parallel replications one can represent $K'_{jm_{jr}\ell+1}$ as

$$K'_{jm_{jr}\ell+1} = \sum_{m=1}^{K'_{j\ell}} I_{[q_{j,r-1},\, q_{jr})}(U_m) \tag{2}$$

where

$$q_{j0} = 0$$

$$q_{jw} = \sum_{r=1}^{w} p_{jm_{jr}} \qquad w=1,\ldots,s_j;\ j=0,1,\ldots$$

and

$$\begin{aligned} I_{[u,v)}(x) &= 1 \qquad && u \le x < v \\ &= 0 && \text{otherwise.} \end{aligned}$$

To ensure that each replication begins with an exit from state a and ends with the first entry into state b , one replaces the original $\{p_{bj};\ j=1,\ldots,s_b\}$ by $p_{bb} = 1$ and sets $p_{bm_{br}} = 0$ for

†The prime superscript is used here for consistency with the notation in Fishman (1981a, 1981b). However, without loss of generality, we take the initial state here as a and the final state as b whereas in the earlier work the initial and final states were 0 and a respectively. This relabeling of initial and final states makes the generality of the approach more apparent to the reader.

$r = 1,\dots,s_b$ and $r \neq b$. If $b = a$, these modifications are made after the first transition.

If one uses the *rotation sampling plan*

$$U_m = U + \frac{m-1}{K'_{j\ell}} \qquad 0 \le U < 1 - \frac{m-1}{K'_{j\ell}} \tag{3}$$

$$= U + \frac{m-1}{K'_{j\ell}} - 1 \qquad 1 - \frac{m-1}{K'_{j\ell}} \le U < 1 ,$$

then the results in Fishman (1981a,b) apply. In particular,

$$\text{var } K'_{ij\ell} \le O(1) \tag{4}$$

and for $S_b = S - b$, the sample number of transitions from i to j is

$$R'_{ijk} = \sum_{\ell=1}^{\infty} K'_{ij\ell} \tag{5a}$$

and the sample mean reward is

$$R'_k = \frac{1}{k} \sum_{\ell=1}^{\infty} \sum_{i\in S_b} \sum_{j\in S} A_{ij} K'_{ij\ell} \quad . \tag{5b}$$

Then one has

$$\text{var } R'_{ijk} \le O((\ln k)^2) \tag{5c}$$

and, if $|A_{ij}| \le O(1)$,

$$k^2 \text{ var } R'_k \le O(\delta^2(\ln k)^4) \qquad \text{as } n \to \infty \tag{5d}$$

$$k^2 \text{ var } R'_k \le O((n\delta \ln k)^2) \qquad n < \infty.$$

Here we speak of $\{A_{ij}\}$ as the reward function. Finally, the number of steps to total absorption

$$T'_k = \min(t\colon K'_{bt} = k)$$

has

$$E\, T'_k = O(\ln k)$$

and

$$\operatorname{var} T'_k \leq O((\ln k)^2)\ . \qquad (6)$$

Note that the sampling scheme (3), when used in (2), preserves the identical and correct probability laws along the sample paths of each of the k replications. An equivalent expression for $K'_{jm_{jr}\ell+1}$, which leads to a computational saving, is:

$$\begin{aligned} K'_{jm_{jr}\ell+1} &= \lfloor Q \rfloor - \lfloor P \rfloor + I_{[\bar{P}, \bar{Q})}(\overline{K'_{j\ell}U}) && \bar{P} \leq \bar{Q} \\ &= \lfloor Q \rfloor - \lfloor P \rfloor - I_{[\bar{Q}, \bar{P})}(\overline{K'_{j\ell}U}) && \bar{P} > \bar{Q} \end{aligned} \qquad (7)$$

where

$$P = K'_{j\ell}\, q_{j,r-1}\ ,$$

$$Q = K'_{j\ell}\, q_{jr}$$

and $\bar{x} = x - \lfloor x \rfloor$. Here the cost of sampling the transitions from state j based on (7), and using the inverse transform method, is $O(c_j) \le O(2\delta + 1)$ and independent of k whereas sampling cost based on (2) is at best $O(k)$.

Let S_k' denote the cost of simulating (1) using (3) in (7). For a countably infinite state space, no more than $(2\delta + 1)\ell$ transient states are occupied prior to transition $\ell+1$ and, therefore, no more than $(2\delta + 1)(\ell^2 + \ell)/2$ sampling events occur through transition ℓ . Since each (i,j) transition has cost $O(2\delta + 1)$, one has $S_k' \le O((\delta T_k')^2)$ so that

$$E\, S_k' \le O((\delta \ln k)^2) \, . \tag{8a}$$

For the finite state case

$$E\, S_k' \le O(n\,\delta \ln k) \, . \tag{8b}$$

These results compare favorably with the case of independent replications taken in series where R_{ijk} and R_k , the sample number of transitions from i to j and the sample mean reward, respectively, have $\operatorname{var} R_{ijk} = O(k)$ and $k \operatorname{var} R_k = O(1)$. Moreover, for simulation cost S_k , one has $O(k) \le E\, S_k \le O(\delta k)$. The lower bound arises when n is small enough to allow storage of all distributions and their *aliases* required by the alias method (Walker 1977) to determine the entered state from each existing state at each transition. The upper bound arises from use of the inverse transform method to determine the paths. See Fishman (1981b) for a more detailed discussion of this cost.

It is also noteworthy that the desirability of k parallel replications with rotation sampling relative to a simulation of k independent replications in series continues to hold when $\{A_{ij}\}$ is not bounded, provided that $\text{var}(A_{ij}\, K'_{ij,\ell+1} | K'_{i\ell}) < \infty$.

2. Potential Uses

This section describes three uses to which the previously described rotation sampling plan can be put with regard to estimation.

First Passage Times

Let

h_ℓ = probability of moving from state a to state b for the first time in exactly ℓ steps

and (9)

$H_\ell = \sum_{i=1}^{\ell} h_i$ = probability of moving from state a to state b for the first time in no more than ℓ steps.

As estimators of h_ℓ and H_ℓ one has, respectively,

$$\hat{h}_\ell = \frac{1}{k} \sum_{j\epsilon S_b} K'_{jb\ell} = \frac{1}{k}(K'_{b\ell} - K'_{b,\ell-1}) \qquad K'_{b0} = 0$$

and (10)

$$\hat{H}_\ell = \frac{1}{k} \sum_{i=1}^{\ell} \sum_{j\epsilon S_b} K'_{jbi} \quad . \qquad \ell = 1,2,\dots \ .$$

Let $S'_{k\ell}$ denote the cost of simulating (1) up to and including step using the rotation sampling plan (3) with (7). Theorem 1 gives several relevant properties for $\hat{h}_\ell$, $\hat{H}_\ell$ and $S'_{k\ell}$.

Theorem 1. For a simulation of k parallel replications of (1) using (3) and (7) one has:

(i) $E\,\hat{h}_\ell = h_\ell$.

(ii) $E\,\hat{H}_\ell = H_\ell$.

(iii) $\operatorname{var} \hat{h}_\ell \leq O((\delta/k)^2)$.

(iv) $\operatorname{var} \hat{H}_\ell \leq O((\delta\ell/k)^2)$.

(v) $E\,S'_{k\ell} \leq O((\delta\ell)^2)$.

Proof. Since the correct probability law continues to prevail on sample paths for each replication, one has $E\,K'_{jb\ell} = k\,h_\ell$ so that (i) and (ii) follow immediately. Since $\operatorname{var} K'_{jb\ell} \leq O(1)$ and no more than 2δ states can transit to state b , it is clear that

$$\operatorname{var}(\textstyle\sum_{j \in S_b} K'_{jb\ell}) \leq O(\delta^2)$$

and that (iii) holds. Since there are exactly ℓ steps to consider, (iv) follows immediately. Here $\hat{H}_\ell$ is a special case of (5b) with $A_{ib} = 1$ $i \in S_b$ and $A_{ij} = 0$ otherwise. Since exactly ℓ steps occur no more than $(2\delta + 1)(\ell^2 + \ell)/2$ trials are necessary each with cost $O(\delta)$. Therefore, (v) obtains.

If one were to perform k independent replications in series, then $\operatorname{var} \hat{h}_\ell = O(1/k)$, $\operatorname{var} \hat{H}_\ell = O(\ell^2/k)$ and $E\,S_k \geq O(k)$, emphasizing the benefit of rotation sampling plan (3) and the conciseness of (7).

Steady-State Distribution

Let

p_j = probability of being in state j at an arbitrarily selected time $j = 0,1,\dots,n$.

As an estimator of p_j one has

$$\hat{p}_j = G_{jk}/G_k \tag{11}$$

where

$$G_{jk} = \frac{1}{k}\sum_{t=1}^{\infty}\sum_{i=0}^{n} K'_{ijt} \qquad j=0,1,\dots,n$$

and (12)

$$G_k = \sum_{j=0}^{n} G_{jk} .$$

Since $\hat{p}_j$ is a *ratio* estimator it has bias

$$E(\hat{p}_j - p_j) \doteq p_j^2 \left[\frac{\text{var } G_k}{E^2 G_k} - \frac{\text{cov}(G_{jk}, G_k)}{E\,G_{jk} \cdot E\,G_k}\right] \tag{13}$$

and variance

$$\text{var } \hat{p}_j \doteq p_j^2 \left[\frac{\text{var } G_k}{E^2\,G_k} - \frac{2\,\text{cov}(G_{jk},G_k)}{E\,G_{jk} \cdot E\,G_k} + \frac{\text{var } G_{jk}}{E^2\,G_{jk}}\right] . \tag{14}$$

From the results of Section 1, one has

$$\operatorname{var} G_{jk} \le O((\delta \ln k/k)^2) \quad \text{and} \quad \operatorname{var} G_k \le O((n \ln k/k)^2)$$

so that both bias and variance benefit from rotation sampling.

Steady-State Probabilities for a Semi-Markov Process

The results presented so far relate directly to a Markov chain. However, their extension to semi-Markov processes is relatively direct for the steady-state probabilities. Let μ_{ij} denote the mean time spent in state i prior to transiting to state j . For the steady-state probabilities, one now has the estimators

$$\tilde{p}_j = G'_{jk}/G'_k \qquad j=0,\dots,n$$

where

$$G'_{jk} = \sum_{t=1}^{\infty} \sum_{i=0}^{n} \mu_{ij} K'_{ijt}$$

and

$$G'_k = \sum_{j=0}^{n} G'_{jk} \quad .$$

Then $\tilde{p}_j$ has approximate bias and variance as in (13) and (14), respectively, with G'_{jk} replacing G_{jk} and G'_k replacing G_k .

3. Implementation

Procedure MC decribes an algorithm that one can use to encode the essential steps for simulating k replications in parallel with initial state a and absorbing state b . In practice, one will want to embellish this procedure to allow for multiple reward functions and multiple independent macroreplications. The latter enable one to estimate variances and covariances of interest.

Figure 1 lists a FORTRAN subprogram called CHAIN that enables one to simulate an $(N + 1)$ state Markov chain with state space on the integers $0,1,\ldots,N$ using the parallel rotation sampling plan (7) for the purpose of computing sample mean rewards RPRIME(1),...,RPRIME(L) for L reward matrices stored in array $A(\cdot)$. Each of I independent *macroreplications* begins with a departure of K correlated *microreplications* from state INITAL and ends with the absorption of all K microreplications in state ABSORB. The purpose of the macroreplications is to facilitate the estimation of the covariance matrix of the L sample mean reward functions. Also, CHAIN estimates the first passage time distribution from INITAL to ABSORB.

Insert Fig. 1 about here.

<u>**Procedure MC**</u>

```
Given: $a, b, k, n, \{s_i; i=0,\ldots,n\}$, $\{m_{ir}: r=1,\ldots,s_i; i=0,\ldots,n\}$,
       $\{p_{i,m_{ir}}: r=1,\ldots,s_i; i=0,\ldots,n\}$ and $\{A_{im_{ir}}; r=1,\ldots,s_i, i=0,\ldots,n\}$.

1.  $i \leftarrow a$.                                  ⎫
2.  $R' \leftarrow 0$.                                 ⎪
3.  $K'_a \leftarrow k$.                               ⎪
4.  For $i=0,\ldots,n$                                 ⎬ Initialize.
        For $r=1,\ldots,s_i$                           ⎪
            $q_{i,m_{ir}} \leftarrow \sum_{\ell=1}^{s_i} p_{i,m_{i\ell}}$.   ⎪
                $K^*_i \leftarrow 0$.                  ⎭
5.  Go to 8.
6.  $i \leftarrow 0$.
7.  If $i=b$ or $K'_i = 0$ go to 23.   (Skip if state is absorbing or empty.)
8.  $r \leftarrow 1$.                                  ⎫
9.  Sample $U$ from $\mathcal{U}(0,1)$.                ⎪
10. $P^* \leftarrow 0$.                                ⎪
11. $\bar{P} \leftarrow 0$.                            ⎪
12. $Q \leftarrow K'_i q_{i,m_{ir}}$.                  ⎬ Rotation Sampling
13. $Q^* \leftarrow \lfloor Q \rfloor$.                ⎪
14. $\bar{Q} \leftarrow Q - Q^*$.                      ⎪
15. $X \leftarrow Q^* - P^* + \frac{1}{2}[\mathrm{sign}(U-\bar{P}) + \mathrm{sign}(\bar{Q}-U)]$.  ⎭
16. $K^*_{m_{ir}} \leftarrow K^*_{m_{ir}} + X$.      (X microreplications move from $i$ to $m_{ir}$.)
17. $R' \leftarrow R' + X A_{i,m_{ir}}$.             (Compute reward.)
18. $P^* \leftarrow Q^*$.
19. $\bar{P} \leftarrow \bar{Q}$.
```

```
20. If $i=b$, $K'_b \leftarrow 0$.     (In case $a = b$.)
21. $r \leftarrow r + 1$.
22. If $r \le s_i$ go to 12.
23. $i \leftarrow i + 1$.
24. If $i \le n$ go to 7.
25. $i \leftarrow 0$.
26. If $i=b$ go to 33.
27. $r \leftarrow 1$.
28. If $m_{ir} = b$, $K'_b \leftarrow K'_b + K^*_b$ and go to 30.     (Arrange for absorptions.)
29. $K'_{m_{ir}} \leftarrow K^*_{m_{ir}}$ .  (Move to transient states.)
30. $K^*_{m_{ir}} \leftarrow 0$.
31. $r \leftarrow r + 1$ .
32. If $r \le s_i$ go to 28.     (Are all moves from $i$ completed?)
33. $i \leftarrow i + 1$ .
34. If $i \le n$ go to 26.
35. If $K'_b < k$ go to 6.   (Are all absorbed?)
36. $R' \leftarrow R'/k$ and deliver $R'$ .
```

Figure 2 contains a partitioned list of the input to CHAIN. Figure 2a displays all arrays and variables for which an initial numerical assignment is necessary at the time at which CHAIN is called. Figure 2b contains all interim working arrays and Fig. 2c lists all output arrays.

Insert Fig. 2 about here.

Note that the arrays A, M and P are one-dimensional in the subprogram whereas their counterparts $\{A_{ij}\}$, $\{m_{ij}\}$ and $\{p_{ij}\}$ are doubly subscripted in Section 1. This reduction leads to a considerable space saving, especially when $\{A_{ij}\}$ $\{m_{ij}\}$ and $\{p_{ij}\}$ are sparse and N is large. In terms of storage space, CHAIN requires $O(4(ALL(L+4) + 2L(L+3) + 4NP + SIZE + 8TT))$ bytes for the arrays listed in Fig. 2. Also, CHAIN has an upper bound on mean execution time of $O(ALL \times I \times L \times \ln K)$.

CHAIN uses the GGUBS random number generator in IMSL (1977) with SEED as the seed or initial random number and returns SIZE uniform deviates on each call of the generator. By choosing the blocking factor SIZE to be large one reduces the frequency of calling GGUBS, thus reducing CPU time. However, the space requirement for the uniform deviates is 4*SIZE bytes. On a computer with limited space, one may select a small SIZE to accomodate the space constraint. More generally, an alternative random number generator can be substituted by GGUBS with little effort.

The input OLD also calls for explanation. After running CHAIN for, say I1 independent macroreplications each of K correlated microreplications, a user may find that the accuracy of the sample mean rewards is too low for intended purposes. By setting OLD = I1 on a second run, choosing a new $I > OLD$, using the original K and restoring XBAR(*) and COV(*,*) in the *driver program*, one can merge the sample output from the first OLD macroreplications with that from the subsequent $I - OLD$ new macroreplications to produce a summary tableau. When this is done, the resulting sample first passage time distribution is based on the last $I - OLD$ macroreplications only.

Macroreplication versus Microreplication

As Section 1 shows, rotation sampling applied to K parallel microreplications produces a covariance matrix whose convergence rate has an upper bound $O((\ln K)^2/K^2)$ on a single macroreplication. Since a method is not yet available for estimating this covariance matrix from a single macroreplication, CHAIN resorts to running I independent macroreplications for the purpose of estimating the covariance matrix of the L sample reward functions. Therefore, the covariance matrix has a convergence rate bounded by $O((\ln K)^2 /IK^2)$ and a run time of $O(I \ln K)$ for fixed ALL and L . Clearly one wants a K substantially larger than I . In the example to be described next $K = 2^{17}$ and $I = 2^3$, but other compromises are equally reasonable.

4. An Example

To illustrate how the CHAIN subprogram works in practice, consider the Markov chain associated with the M/M/1 queueing problem with finite capacity n . In particular, the *state* of the chain denotes the number of customers in the system. In this queueing problem interarrival times are i.i.d. from the exponential distribution with rate λ, service times are i.i.d. from the exponential distribution with rate $\omega > \lambda$ and there is a single server. For the corresponding Markov chain these specifications imply

$$\begin{aligned} p_{01} &= 1 & & \\ p_{i,i-1} &= \frac{\omega}{\lambda+\omega} & & i=1,\ldots,n \\ p_{i,i+1} &= \frac{\lambda}{\lambda+\omega} & & i=1,\ldots,n-1 \\ p_{n,n} &= \frac{\lambda}{\lambda+\omega} & & \end{aligned}$$

with all other transition probabilities being zero.

As objectives consider the estimation of

W_1 = mean number in system.

W_2 = probability of one customer in the system.

W_3 = probability of two customers in the system.

W_4 = first passage time probability mass function for the Markov chain from the empty and idle state back to that state.

W_5 = distribution function associated with W_4 .

Let $\{A_{ij}(\ell)\}$ denote reward matrix ℓ and $R_k'(\ell)$ denote sample mean reward function ℓ based on using $\{A_{ij}(\ell)\}$ in (5b). Then one estimates W_1, W_2 and W_3 by

$$\hat{W}_1 = R_k'(1)/R_k'(2)$$
$$\hat{W}_2 = R_k'(3)/R_k'(2)$$
$$\hat{W}_3 = R_k'(4)/R_k'(2)$$

where

$$A_{i,i+1}(1) = \frac{i+1}{\lambda+\omega} \qquad i=0,\dots,n-1$$
$$A_{i,i-1}(1) = \frac{i-1}{\lambda+\omega} \qquad i=1,\dots,n$$
$$A_{n,n}(1) = \frac{n}{\lambda+\omega}$$
$$A_{10}(2) = \frac{1}{\lambda}$$
$$A_{i,i+1}(2) = \frac{1}{\lambda+\omega} \qquad i=0,\dots,n-1$$
$$A_{i,i-1}(2) = \frac{1}{\lambda+\omega} \qquad i=2,\dots,n$$
$$A_{n,n}(2) = \frac{1}{\lambda+\omega}$$
$$A_{10}(3) = \frac{\omega}{\lambda+\omega}$$
$$A_{12}(3) = \frac{\lambda}{\lambda+\omega}$$
$$A_{21}(4) = \frac{\omega}{\lambda+\omega}$$
$$A_{23}(4) = \frac{\lambda}{\lambda+\omega} \quad .$$

For W_4 and W_5 we use the estimators in (10). These are computed automatically by CHAIN.

Figure 3 shows a driver program designed to initialize all relevant parameters and call CHAIN for this problem. Here LAM and W correspond to λ and ω

Insert Fig. 3 about here.

respectively. As input we set

LAM = 0.9

W = 1.0

N = 49

INITAL = 0

ABSORB = 0

I = 2^3 = 8

K = 2^{17} = 131072

L = 4

SEED = 1234567

SIZE = 40000

TT = 50 .

Figure 4 shows the output of CHAIN for this problem. The ratio estimators W_1, W_2 and W_3 together with their biases and variances can

Insert Fig. 4 about here.

be computed by hand or by a subsequent subroutine that uses XBAR and COV as input together with the formulae

$$E\left(\frac{Y}{X} - \frac{EX}{EX}\right) \doteq \frac{EY}{EX}\left[\frac{\text{var } X}{E^2 X} - \frac{\text{cov}(X,Y)}{EX \cdot EY}\right]$$

and

$$\text{var}(Y/X) \doteq \frac{E^2 Y}{E^2 X}\left[\frac{\text{var } X}{E^2 X} - \frac{2\,\text{cov}(X,Y)}{EX \cdot EY} + \frac{\text{var } Y}{E^2 Y}\right] .$$

We have chosen to do them by hand. They are

$$\hat{W}_1 = 8.7396 \qquad \hat{W}_2 = .0905 \qquad \hat{W}_3 = .0814$$

$$E(\hat{W}_1 - W_1) \doteq -.1037 \times 10^{-4} \quad E(\hat{W}_2 - W_2) \doteq .4304 \times 10^{-7} \quad E(\hat{W}_3 - W_3) \doteq .3735 \times 10^{-7}$$

$$\text{var } \hat{W}_1 \doteq .2646 \times 10^{-3} \quad \text{var } \hat{W}_2 \doteq .3780 \times 10^{-8} \quad \text{var } \hat{W}_3 \doteq .2936 \times 10^{-8} .$$

From Gross and Harris (1974, Sec. 2.5) one has

$$W_1 = 8.7410 \qquad W_2 = .0905 \qquad W_3 = .0814 ,$$

which confirm the performance of CHAIN.

All computing was performed on an IBM 370/155 computer. For FORTRAN G (level 21), the CHAIN Program requires 8592 bytes of space. For FORTRAN H (level 21.8) it requires 7380 bytes.

5. Restarting the Simulation

Situations will occur in which a user of CHAIN does not have a good a priori estimate of the running time or the statistical reliability to be expected from a specified set of I_1 macroreplications each of K_1 microreplications. At least two alternatives exist for dealing with this case. A user may make a preliminary run and determine that I_2 additional macroreplications, each of $K_2 = K_1$ microreplications, are necessary to achieve the desired accuracy within budget. CHAIN is designed to accommodate this alternative and, provided that the sample means and covariances accumulated on the first I_1 macroreplications are restored prior to collecting the additional I_2 macroreplications, the routine prints the global sample means and covariances at the end of all $I_1 + I_2$ macroreplications.

The second alternative arises when on the basis of the first run a user determines that I_2 macroreplications each of $K_2 \neq K_1$ microreplications is the most desirable way of achieving the desired accuracy. Here the restoration feature of CHAIN does not apply. Let $\underset{\sim}{\bar{X}}$ and $\underset{\sim}{\Sigma}$ be the sample mean vector and sample covariance matrix of $\underset{\sim}{\bar{X}}$ obtained on the first run with I_1 and K_1. Let $\underset{\sim}{\bar{Y}}$ and $\underset{\sim}{\Omega}$ denote the corresponding quantities on the second run with I_2 and K_2. Then the overall sample mean vector is

$$\underset{\sim}{\bar{Z}} = \frac{1}{(I_1 + I_2)} (I_1 \underset{\sim}{\bar{X}} + I_2 \underset{\sim}{\bar{Y}})$$

and its sample covariance matrix is

$$\underset{\sim}{\Gamma} = \frac{1}{(I_1 + I_2)^2} (I_1^2 \underset{\sim}{\Sigma} + I_2^2 \underset{\sim}{\Omega}) .$$

Although the computation of $\underset{\sim}{\Sigma}$ and $\underset{\sim}{\Gamma}$ are not features of CHAIN one can easily add them if desired. However, as in the case of the ratio estimators in Section 4, the relative importance of this feature will differ from user to user.

References

1. Fishman, G. S. and B. D. Huang (1980). "Antithetic Variates Revisited," TR 80-4, Curriculum in Operations Research & Systems Analysis, University of North Carolina at Chapel Hill.
2. Fishman, G. S. (1981a)."Accelerated Accuracy in the Simulation of Markov Chains," TR 81-1, Curriculum in Operations Research & Systems Analysis, University of North Carolina at Chapel Hill.
3. Fishman, G. S. (1981b). "Accelerated Convergence in the Simulation of Countably Infinite State Markov Chains," TR 81-4, Curriculum in Operations Research & Systems Analysis, University of North Carolina at Chapel Hill.
4. Gross, D. and C. M. Harris (1974). Fundamentals of Queueing Theory, John Wiley and Sons.
5. International Mathematical and Statistical Libraries, Inc. (1977). IMSL Library, Houston, Texas.
6. Walker, Alistair (1977). "An Efficient Method of Generating Discrete Random Variables with General Distributions," ACM Transactions on Mathematical Software, 3, 253-6.

```
      SUBROUTINE CHAIN(K,OLD,I,NP,INITAL,ABSORB,S,M,SUMS,ALL,P,
     1                KPRIME,KSTAR,Q,L,ASIZE,A,RPRIME,XBAR,COV,COEFF,
     2                TT,FRBAR,FRVAR,CUMBAR,CUMVAR,SEED,SIZE,V)
C *** PROGRAM FOR SIMULATION OF MARKOV CHAIN WITH N+1 STATES
C *** USING ROTATION SAMPLING.

      INTEGER NP,ABSORB,ALL,B,COUNT,I,INITAL,ISEED,IT,J,KPRIME(NP),
     1        KSTAR(NP),L,LA,LI,LIJ,K,LL,LM,R,M(ALL),N,OLD,OMEGA,X,
     2        PSTAR,QSTAR,OMEGA,SEED,SIZE,S(NP),SJ,SK,SS,SUMS(NP),TT,
     3        ASIZE
      REAL*4  A(ASIZE),FRCV,CUMCV,V(SIZE)
      REAL*8  C,CC,COEFF(L),COV(L,L),P(ALL),PP,PBAR,Q(ALL),QQ,QBAR,
     1        RPRIME(L),RSEED,XBAR(L),U,FREQ,CUM,FRVAR(TT),FRBAR(TT),
     2        CUMBAR(TT),CUMVAR(TT),FF,FFVAR,REG

C *** DESCRIPTION OF VARIABLES -
C ***
C *** A(*)      = REWARD MATRIX
C *** ABSORB    = ABSORBING STATE
C *** ALL       = SUM OF S(J) FOR ALL J
C *** ASIZE     = SIZE OF A ARRAY
C *** B         = TEST VARIABLE
C *** C         = ((LI-1)/LI)
C *** CC        = ((LI-OLD-1)/(LI-OLD))
C *** COEFF(*)  = COEFFICIENTS OF VARIATION
C *** COUNT     = UNIFORM DEVIATE COUNTER
C *** COV(*,*)  = COVARIANCE MATRIX
C *** CUM       = ACCUMULATION OF FREQ
C *** CUMBAR(*) = MEAN OF CUM
C *** CUMCV     = COEFFICIENT OF VARIATION FOR CUM
C *** CUMVAR(*) = VARIANCE OF CUM
C *** FF        = SAMPLE MEAN FOR FIRST PASSAGE
C *** FFVAR     = SAMPLE VARIANCE FOR FIRST PASSAGE
C *** FRBAR(*)  = MEAN OF FREQ
C *** FRCV      = COEFFICIENT OF VARIATION FOR FREQ
C *** FREQ      = NUMBER OF NEW TRANSITIONS INTO ABSORBING STATE
C *** FRVAR(*)  = VARIANCE OF FREQ
C *** I         = DESIRED NUMBER OF MACROREPLICATIONS
C *** INITAL    = INITIAL STATE
C *** ISEED     = INITIAL SEED
C *** IT        = TRANSITION COUNTER
C *** J         = INDEX FOR STATES
C *** K         = NUMBER OF PARALLEL MICROREPLICATIONS
C *** KPRIME(*) = NUMBER IN STATE AT END OF TRANSITION
C *** KSTAR(*)  = NEXT KPRIME(*)
C *** L         = NUMBER OF DESCRIPTORS TO BE ESTIMATED
C *** LA        = LL + 1
C *** LI        = INDEX FOR I
C *** LIJ       = 1 + OLD
C *** LJ        = INDEX FOR STATES
```

Fig. 1. CHAIN Subroutine

```
C *** LL        = INDEX FOR L
C *** LB        = INDEX FOR L
C *** M(*)      = STATES THAT CAN BE ENTERED FROM J-1
C *** N         = NUMBER OF HIGHEST STATE
C *** NL        = NUMBER OF STATES (N+1)
C *** OLD       = NUMBER OF MACROREPLICATIONS ALREADY PERFORMED (OLD<I)
C *** OMEGA     = TEST VARIABLE
C *** P(*)      = TRANSITION MATRIX
C *** PP        = QQ FROM TIME BEFORE
C *** PBAR      = FRACTIONAL PART OF PP
C *** PSTAR     = INTEGER PART OF PP
C *** Q(*)      = FROM P
C *** QQ        = FROM Q AND K
C *** QBAR      = FRACTIONAL PART OF QQ
C *** QSTAR     = INTEGER PART OF QQ
C *** R         = INDEX FOR STATES
C *** REG       = ACCUMULATED FREQ FOR TRANSITIONS GREATER THAN TT-1
C *** RPRIME(*)= CUMULATIVE REWARD FOR ESTIMATORS
C *** RSEED     = RANDOM GENERATOR SEED, REAL VALUED
C *** SEED      = RANDOM GENERATOR SEED
C *** SIZE      = BLOCK SIZE FOR RANDOM NUMBER GENERATOR
C *** S(J)      = NUMBER OF STATES THAT CAN BE ENTERED FROM J-1
C *** SJ        = S(J)
C *** SK        = SUMS(J)
C *** SS        = MIN (NUMBER OF TRANSITIONS, TT)
C *** SUMS(J)   = SUM OF S(1) THRU S(J-1), AND SUMS(1)=0
C *** TT        = MAXIMUM NUMBER OF TRANSITIONS TO LOOK AT
C *** U         = CURRENT UNIFORM DEVIATE
C *** V(*)      = UNIFORM DEVIATE ARRAY
C *** X         = NUMBER OF TRANSITIONS IN CURRENT MOVE
C *** XBAR(*)   = MEAN MATRIX

C *** INITIALIZE VARIABLES.

      PP    = 0.0D0
      PPVAR = 0.0D0
      SS    = 0
      ISEED = SEED
      RSEED = SEED
      COUNT = SIZE
      DO 80 J=1,TT
         FRVAR(J)  = 0.0D0
         FRBAR(J)  = 0.0D0
         CUMVAR(J) = 0.0D0
80       CUMBAR(J) = 0.0D0
      DO 100 J=1,NP
         SK        = SUMS(J)
         Q(SK+1)   = P(SK+1)
         IF (S(J).LT.2) GO TO 100
         SJ        = S(J)
```

Fig. 1 (*Continued*)

```
         DO  90 R=2,SJ
 90         Q(SK+R)  =  Q(SK+R-1)+P(SK+R)
         Q(SK+SJ)    =  1.0000D0
100         CONTINUE
C *** CHECK IF XBAR AND COV ARE RESTORED.
      IF (OLD.GT.0) GO TO 115
      DO 110 LL=1,L
         XBAR(LL)    = 0.0D0
         COEFF(LL)   = 0.0D0
         DO 110 LM=1,L
110         COV(LL,LM)  = 0.0D0
115   LIJ = 1 + OLD
      DO 118 LL=1,L
         XBAR(LL) = XBAR(LL)*OLD
         DO 118 LM=1,L
118         COV(LL,LM) = COV(LL,LM)*OLD

C *** START MAIN LOOP.
      DO 540 LI=LIJ,I
C *** INITIALIZE VARIABLES FOR THIS REPLICATION.
         B      = 0
         C      = (LI-1.0D0)/LI
         CC     = (LI-OLD-1.0D0)/(LI-OLD)
         CUM    = 0.0D0
         REG    = 0.0D0
         IT     = 1
         OMEGA  = 0
         DO 120 LL=1,L
120         RPRIME(LL) = 0.0D0
         DO 130 J=1,NP
            KSTAR(J) = 0
130         KPRIME(J)= 0
C *** START THIS REPLICATION WITH ALL MICROREPLICATIONS IN INITIAL STATE
         J          = INITAL + 1
         KPRIME(J)= K
         KSTAR(J) = K
C *** LOOK AT INITIAL STATE.
         J       = INITAL + 1
         GO TO 400
C *** LOOK AT THE NEXT STATE.
300      J       = J+1
C *** SKIP THE ABSORBING STATE.
         IF (J.EQ.ABSORB+1) J = J+1
C *** IF NONE IN THIS STATE, LOOK AT THE NEXT STATE.
400      IF (KPRIME(J).EQ.0) GO TO 300
C *** FIND THE NUMBER OF STATES THAT CAN BE ENTERED FROM STATE J-1.
         SJ      = S(J)
C *** START WITH THE FIRST STATE THAT CAN BE ENTERED FROM STATE J-1.
         R     = 1
C *** POINT TO THE NEXT DEVIATE TO USE.
```

Fig. 1 (*Continued*)

```
      COUNT = COUNT + 1
C *** GET NEW ARRAY OF DEVIATES IF NEEDED.
      IF (COUNT.LE.SIZE) GO TO 420
      COUNT = 1
C *** CALL RANDOM(SEED,V,SIZE)
      CALL GGUBS(RSEED,SIZE,V)
C *** GET THE DEVIATE POINTED TO BY COUNT.
C *** TRANSFORM DEVIATE.
420   U      = DBLE(AMOD(KPRIME(J)*V(COUNT),1.))
C     WRITE(3,2040) U,V(COUNT)
C *** TRANSFER ALL OUT OF THIS STATE FOR TRANSITIONS.
      KSTAR(J) = KSTAR(J) - KPRIME(J)
C *** INITIALIZE QQ, QSTAR, AND QBAR.
      QQ    = 0.0
      QSTAR = 0
      QBAR  = 0.0
C *** SAVE THE LAST OCCURRENCE OF QQ, QSTAR, AND QBAR.
500   PP    = QQ
      PSTAR = QSTAR
      PBAR  = QBAR
C *** COMPUTE NEW VALUES FOR QQ, QSTAR, AND QBAR.
      QQ    = KPRIME(J)*Q(SUMS(J)+R)
      QSTAR = IDINT(QQ)
      QBAR  = DMOD(QQ,1.0D0)
C *** FIND THE # OF TRANSITIONS TO THE R-TH STATE THAT CAN BE ENTERED.
C *** FROM J-1.
      X     = (QSTAR-PSTAR)+.5*(DSIGN(1.0D0,U-PBAR)+
     1                        DSIGN(1.0D0,QBAR-U))
C *** ADD THESE TRANSITIONS TO THE ENTERED STATE OCCUPANCY VECTOR.
      KSTAR(M(SUMS(J)+R)+1) = KSTAR(M(SUMS(J)+R)+1)+X
C *** FIND THE NUMBER OF TRANSITIONS MADE SO FAR FOR THIS STATE.
      B     = B+X
C *** ACCUMULATE THE REWARDS FOR THIS STATE TRANSITION.
      DO 510 LL=1,L
         IF (A(ALL*(LL-1)+SUMS(J)+R).EQ.0.0.OR.X.EQ.0) GO TO 510
         RPRIME(LL) = RPRIME(LL)+A(ALL*(LL-1)+SUMS(J)+R)*X
510      CONTINUE
C *** GO TO THE NEXT STATE THAT CAN BE REACHED FROM J-1.
      R     = R+1
C *** IF NOT ALL TRANSITIONS WERE MADE FOR THIS STATE, TRY AGAIN.
      IF (B.LT.KPRIME(J)) GO TO 500
C *** ACCUMULATE THE NUMBER OF TRANSITIONS MADE SO FAR FOR ALL STATES.
      OMEGA = OMEGA+B
C *** CLEAR THE NUMBER OF TRANSITIONS FOR THE STATE COUNTER.
      B     = 0
C *** IF NOT ALL TRANSITIONS WERE MADE YET, TRY AGAIN.
      IF (OMEGA.LT.K-KPRIME(ABSORB+1)) GO TO 300
      OMEGA = 0
C *** ACCUMULATE FRBAR, FRVAR, CUMBAR AND CUMVAR.
      FREQ = KSTAR(ABSORB+1)-KPRIME(ABSORB+1)
```

Fig. 1 *(Continued)*

```
         IF (IT.EQ.1) FREQ = KSTAR(ABSORB+1)
         FF         = FF + IT*FREQ
         FFVAR      = FFVAR + IT**2*FREQ
         CUM  = CUM + FREQ
         LL   = MIN0(TT,IT)
         IF (IT.LT.TT) GO TO 515
         REG  = REG + FREQ
         IF (KSTAR(ABSORB+1).LT.K) GO TO 522
         FREQ = REG
515      FRBAR(LL)  = FRBAR(LL) + FREQ
         CUMBAR(LL) = CUMBAR(LL) + CUM
         FRVAR(LL)  = FRVAR(LL) + FREQ**2
         CUMVAR(LL) = CUMVAR(LL) + CUM**2
C *** RESET THE STATE OCCUPANCY VECTOR.
522      DO 525 LJ=1,NP
            KPRIME(LJ) = KSTAR(LJ)
            SJ    = S(LJ)
            DO 525 R=1,SJ
525            KPRIME(M(SUMS(LJ)+R)+1) = KSTAR(M(SUMS(LJ)+R)+1)
C *** START NEXT TRANSITION IN STATE 0.
         J  = 0
C *** COMPUTE THE NUMBER OF TRANSITIONS MADE SO FAR.
         IT = IT + 1
C *** IF NOT ALL ABSORBED, TRY AGAIN.
         IF (KPRIME(ABSORB+1).LT.K) GO TO 300
C *** RECURSIVE COMPUTATIONS.
         IF (LL.GE.TT) GO TO 529
C *** ACCUMULATE CUMBAR AND CUMVAR FOR # OF TRANSITION STEPS > TT-1.
         LL = LL + 1
         DO 527 LM=LL,TT
            CUMVAR(LM) = CUMVAR(LM) + CUM**2
527         CUMBAR(LM) = CUMBAR(LM) + CUM
C *** COMPUTE THE COVARIANCE MATRIX RECURSIVELY.
529      IF (LI.EQ.1) GO TO 535
         DO 530 LL=1,L
            DO 530 LM=1,L
530            COV(LL,LM)=((LI-2)*COV(LL,LM)+(KPRIME(LL)/K-XBAR(LL))*
     1                    (KPRIME(LM)/K-XBAR(LM))*C)/(LI-1)
C *** COMPUTE SS.
535      SS = MIN0(TT,MAX0(SS,IT-1))
C *** COMPUTE THE SAMPLE MEAN VECTOR RECURSIVELY.
         DO 540 LL=1,L
            XBAR(LL) = C*XBAR(LL)+(1.0D0-C)*KPRIME(LL)/K
540         CONTINUE

C *** END MAIN LOOP, COMPLETED I MACROREPLICATIONS.
C ***
C *** COMPUTE FRBAR, FRVAR, CUMBAR, AND CUMVAR DIRECTLY.
      IF (I.NE.1) REG = (K**2.0D0)*(I**2.0D0)*(I-1.0D0)
      DO 545 LL=1,TT
```

Fig. 1 (*Continued*)

```
         IF (I.EQ.1) GO TO 544
         PRVAR(LL)   =  (I*PRVAR(LL)-PRBAR(LL)**2)/REG
         CUMVAR(LL)  =  (I*CUMVAR(LL)-CUMBAR(LL)**2)/REG
544      CUMBAR(LL)  =  CUMBAR(LL)/(K*I)
545      PRBAR(LL)   =  PRBAR(LL)/(K*I)

      DO 550 LL=1,L
         DO 550 LM=1,L
550         COV(LL,LM)  =  COV(LL,LM)/L
      DO 570 LL=1,L
         IF (XBAR(LL).NE.0) COEFF(LL)  =  DSQRT(COV(LL,LL))/XBAR(LL)
         IF (LL.EQ.L)  GO TO 570
         LA  =  LL+1
         DO 560 LM=LA,L
            IF  (COV(LL,LL)*COV(LM,LM).GT.0)
     1         COV(LM,LL)=COV(LL,LM)/DSQRT(COV(LL,LL)*COV(LM,LM))
560         CONTINUE
570      CONTINUE

C *** PRINT RESULTS.

      SEED  =  MSEED
      WRITE (3,1000)  NP,INITAL,ABSORB,ALL,L,K,I,ISEED,SEED,SIZE
1000  FORMAT(1H1,'RESULTS OF ROTATION SAMPLING FOR A MARKOV CHAIN',///,
     1'   NO. OF STATES                              =',I10//,
     2'   INITIAL STATE                              =',I10//,
     3'   ABSORBING STATE                            =',I10//,
     4'   TOTAL NO. OF (I,J) PAIRS                   =',I10//,
     5'   NO. OF DESCRIPTORS                         =',I10//,
     6'   NO. OF CORRELATED MICROREPLICATIONS        =',I10//,
     7'   NO. OF INDEPENDENT MACROREPLICATIONS       =',I10//,
     8'   INITIAL SEED                               =',I10//,
     9'   FINAL SEED                                 =',I10//,
     C'   BLOCKING FACTOR                            =',I10/)
      REWIND 11
      WRITE (3,2000)
      WRITE (3,2030)
      WRITE (3,2040) (XBAR(LL),LL=1,L)
      WRITE (3,2010)
      WRITE (3,2030)
      WRITE (3,2040) (COEFF(LL),LL=1,L)
      WRITE (3,2020)
      WRITE (3,2030)
      DO 580 LL=1,L
         WRITE (3,2045) LL, (COV(LL,LM),LM=1,L)
580   CONTINUE
2000  FORMAT(//,' SAMPLE MEAN VECTOR',/,
     1          ' ******************',/)
2010  FORMAT(//,' SAMPLE COEFFICIENTS OF VARIATION',/,
     1          ' ********************************',/)
```

Fig. 1 *(Continued)*

```
2020  FORMAT(//,' SAMPLE COVARIANCE/CORRELATION MATRIX',/,
     1          ' ************************************',/)
2030  FORMAT(9X,'1',14X,'2',14X,'3',14X,'4',14X,'5',14X,'6',14X,'7',
     1       14X,'8',14X,'9',//)
2040  FORMAT(2X,9(D15.8)/)
2045  FORMAT(I2,9(D15.8)/)
      FF         = FF/(K*I)
      FFVAR      = FFVAR/(K*I) - FF**2
      DO 620 LL=1,SS
         PRCV  = 0.0
         CUMCV = 0.0
         IF (PRBAR(LL).NE.0.0D0) PRCV=DSQRT(PRVAR(LL))/PRBAR(LL)
         IF (CUMBAR(LL).NE.0.0D0) CUMCV=DSQRT(CUMVAR(LL))/CUMBAR(LL)
         IF (MOD(LL,50).NE.1) GO TO 620
         WRITE (3,3000)
         IF (LL.GT.1) GO TO 610
         WRITE (3,3010) FF,FFVAR
610      WRITE (3,3020)
620      WRITE (3,3030) LL,PRBAR(LL),PRVAR(LL),PRCV,CUMBAR(LL),
     1                  CUMVAR(LL),CUMCV
      IF (OLD.EQ.0) RETURN
      LL = I - OLD
      WRITE (3,4000) LL
3000  FORMAT(1H1,'FIRST PASSAGE TIME (T) DISTRIBUTION',/,
     1         ' ***********************************')
3010  FORMAT(/,1X,'SAMPLE MEAN(T) =',D15.8,5X,
     1          'SAMPLE VARIANCE(T) =',D15.8)
3020  FORMAT(/,33X,'MASS FUNCTION',42X,'DISTRIBUTION FUNCTION',//,
     1                             35X,'VARIANCE',10X,'COEFFICIENT',
     2                             31X,'VARIANCE',10X,'COEFFICIENT',/,
     3    '    I',10X,'PR(T=I)', 11X,'OF PR(T=I) ',9X,'OF VARIATION',
     4           10X,'PR(T<=I)',10X,'OF PR(T<=I)',9X,'OF VARIATION',
     5  /,'   ---',9X,'-------', 11X,'-----------',9X,'------------',
     6           10X,'--------',10X,'-----------',9X,'------------')
3030  FORMAT(' ',I5,6(5X,D15.8))
4000  FORMAT(/,' FIRST PASSAGE DISTRIBUTION IS BASED ON THE LAST ',
     1       I5,' MACROREPLICATIONS.')
      RETURN
      END
```

Fig. 1 (*Continued*)

Fig. 2 Input to CHAIN Subprogram

(a) Data Input

VARIABLE	TYPE	DESCRIPTION
A	REAL*4(ASIZE)	A(ALL*(LL-1) + SUMS(J) + R) = reward received when a replication jumps from state J - 1 to state M(SUMS(J) + R) for R = 1,...,S(J) for reward function LL = 1,2,...,L
ABSORB	INTEGER	Absorbing state 0 ≤ ABSORB ≤ N + 1
ALL	INTEGER	Total number of arcs = SUMS(NP) + S(NP)
ASIZE	INTEGER	ALL*L
I	INTEGER	Desired number of independent macroreplications
INITAL	INTEGER	Initial state 0 ≤ INITAL ≤ N + 1
K	INTEGER	Number of parallel microreplications per macroreplication
L	INTEGER	Total number of reward functions
M	INTEGER(N+1)	M(SUM(J) + LR) = LRth of S(J) states to which a replication can move from state J - 1 LR = 1,...,S(J)
NP	INTEGER	NP = N + 1 = total number of states
OLD	INTEGER	If OLD = 0 simulation proceeds to run I macroreplications If OLD > 0 simulation proceeds to run I - OLD additional macroreplications
P	REAL*4(ALL)	P(SUM(J) + LR) = probability of moving from state J - 1 to M(SUM(J) + LR) LR = 1,...,S(J)
S	INTEGER(N+1)	S(J) = number of states that can be entered from state J - 1
SEED	INTEGER	Initial value for random number generator
SIZE	INTEGER	Each call to the random number generator returns a block of SIZE uniform deviates

(continued)

Fig. 2 (*Continued*)

(a)

VARIABLE	TYPE	DESCRIPTION
SUMS	INTEGER(NP)	SUMS(J) = 0 J = 1 $= \sum_{I=1}^{J-1} S(I)$ J = 2,...,NP
TT	INTEGER	Number of cells in sample first passage time distribution. Last cell estimates probability of absorption at time ≥ TT

(*b*) Working Arrays

VARIABLE	TYPE	DESCRIPTION
KPRIME	INTEGER(NP)	Distribution of microreplications by state at end of a transition
KSTAR	INTEGER(NP)	Distribution of microreplications by state at beginning of a transition
Q	REAL*8(ALL)	Q(SUM(J) + LR) = probability of moving from state J - 1 to state M(SUM(J) + 1), M(SUM(J) + 2),... or M(SUM(J) + LR) LR = 1,...,S(J)
RPRIME	REAL*8(L)	Accumulated rewards for L reward functions
V	REAL*4(SIZE)	Space to store uniform deviates

(continued)

Fig. 2 (*Continued*)

(c) Arrays Used to Summarize Data on I Macroreplications

VARIABLE	TYPE	DESCRIPTION
COEFF	REAL*8(L)	COEFF(J1) = sample coefficient of variation of XBAR(J1) J1=1,...,L
COV	REAL*8(L,L)	At completion COV(J1, J2) contains the sample covariance of XBAR(J1) and XBAR(J2) for J2=J1,...,L and J1=1,...,L and COV(J2,J1) contains the sample correlation between XBAR(J1) and XBAR(J2) for J1=1,...,J2-1 and J2=2,...,L .
CUMBAR	REAL*8(TT)	CUMBAR(J1) = sample probability that absorption occurs on a step ≤ J1 for J1=1,...,TT - 1; CUMBAR(TT) = 1
CUMVAR	REAL*8(TT)	CUMVAR(J1) = Sample variance of CUMBAR(J1) J1=1,...,TT
FF	REAL*8	Sample mean of first passage time.
FFVAR	REAL*8	Sample variance of first passage time.
FRBAR	REAL*8(TT)	FRBAR(J1) = sample absorption probability at step J1 for J1=1,...,TT - 1 $\text{FRBAR(TT)} = 1 - \sum_{J1=1}^{TT-1} \text{FRBAR(J1)}$
FRVAR	REAL*8(TT)	FRVAR(J1) = sample variance of FRBAR(J1) J1=1,...,TT
XBAR	REAL*8(L)	XBAR(J1) = the sample mean reward for reward function J1=1,...,L

```
      INTEGER I,J,KPRIME(50),KSTAR(50),L,K,LL,M(100),N,NP,S(50),
     1        SUMS(50),ALL,INITAL,ABSORB,SIZE,OLD,SEED,TT,ASIZE
      REAL*4  A(500),LAM,V(40000),W
      REAL*8  COEFF(10),COV(10,10),Q(100),PRIME(10),XBAR(10),
     1        P(100),PRVAR(100),PRBAR(100),CUMBAR(100),CUMVAR(100)

C *** INITIALIZE VALUES.

      READ (1,1000) N,L,K,LAM,W,I,ABSORB,INITAL,SIZE,TT
1000  FORMAT (I5,/,I3,/,I6,/,F4.1,/,F4.1,/,I5,/,I5,/,I5,/,I5,/,I5)
      WRITE(3,1005) N,L,K,LAM,W,I,ABSORB,INITAL,SIZE,TT
1005  FORMAT (' N=',I5,' L=',I3,' K=',I6,
     1        ' LAM=',F4.1,' W=',F4.1,' I=',I5,' ABSORB=',I5,
     2        ' INITAL=',I5,' SIZE=',I5,' TT=',I5)
      NP = N + 1
      READ (11,1020) SEED
1020  FORMAT(I10)
C *** FOR EACH STATE DETERMINE THE NUMBER OF STATES THAT CAN BE ENTERED.
      S(1) = 1
      DO 80 J=2,NP
80       S(J) = 2
      WRITE(3,1015) (S(J),J=1,NP)
1015  FORMAT (' S(J)      =',20I5)
      SUMS(1) = 0
      ALL   = S(1)
      DO 90 J=2,NP
         SUMS(J) = ALL
90       ALL   = ALL + S(J)
C *** THE NEXT INITIALIZATIONS MAY BE RUN SPECIFIC.
      OLD = 0
      DO 100 J=2,NP
C ***    COMPUTE TRANSITION PROBABILITIES.
         P(SUMS(J)+1) =   W/(LAM+W)
         P(SUMS(J)+2) = LAM/(LAM+W)
100      CONTINUE
      P(1)          = 1.0
      WRITE (3,1016) (P(LK),LK=1,ALL)
1016  FORMAT (' P(*)      =',20F5.2)
C *** FOR EACH STATE J DETERMINE STATES THAT CAN BE ENTERED.
      DO 400 J=2,N
         M(SUMS(J)+1) = J-2
400      M(SUMS(J)+2) = J
      M(1)             = 1
      M(SUMS(NP)+1) = N-1
      M(SUMS(NP)+2) = N
      WRITE (3,1017) (M(LL),LL=1,ALL)
1017  FORMAT (' M(*)      =',20I5)
C *** COMPUTE REWARD VECTOR.
      DO 500 J=1,NP
         A(ALL*2+SUMS(J)+1) = 0.0
```

Fig. 3 Driver Routine for Example

```
          A(ALL*2+SUMS(J)+2)  = 0.0
          A(ALL*3+SUMS(J)+1)  = 0.0
          A(ALL*3+SUMS(J)+2)  = 0.0
          A(SUMS(J)+1)        = M(SUMS(J)+1)/(LAM+W)
          A(SUMS(J)+2)        = M(SUMS(J)+2)/(LAM+W)
          A(ALL+SUMS(J)+1)    = 1/(LAM+W)
500       A(ALL+SUMS(J)+2)    = 1/(LAM+W)
      A(ALL+SUMS(2)+1)        = 1/LAM
      A(ALL*2+1)              = 1/(LAM+W)
      A(ALL*2+SUMS(3)+1)      = 1/(LAM+W)
      A(ALL*3+SUMS(2)+2)      = 1/(LAM+W)
      A(ALL*3+SUMS(4)+1)      = 1/(LAM+W)
      DO 550 LL=1,L
      WRITE (3,1018) (A(ALL*(LL-1)+LR),LR=1,ALL)
1018  FORMAT (' A(*,LL)  =',20F5.2)
550   CONTINUE
      ASIZE = ALL*L
C *** CALL PARALLEL SIMULATION PROGRAM.
      CALL CHAIN(K,OLD,I,NP,INITAL,ABSORB,S,M,SUMS,ALL,P,KPRIME,
     1          KSTAR,Q,L,ASIZE,A,PRIME,XBAR,COV,COEFF,TT,PRBAR,
     2          PRVAR,CUMBAR,CUMVAR,SEED,SIZE,V)
      STOP
      END
```

Fig. 3 (*Continued*)

```
N=    49 L=  4 K=131072 LAM= 0.9 V= 1.0 I=     8 ABSORB=    0 INITAL=    0 SIZE=40000 TT=   40
S(J)      =   1    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
S(J)      =   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
S(J)      =   2    2    2    2    2    2    2    2    2    2
P(*)      = 1.00 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53
P(*)      = 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53
P(*)      = 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53
P(*)      = 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53
P(*)      = 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47 0.53 0.47
N(*)      =    1    0    2    1    3    2    4    3    5    4    6    5    7    6    8    7    9    8   10    9
N(*)      =   11   10   12   11   13   12   14   13   15   14   16   15   17   16   18   17   19   18   20   19
N(*)      =   21   20   22   21   23   22   24   23   25   24   26   25   27   26   28   27   29   28   30   29
N(*)      =   31   30   32   31   33   32   34   33   35   34   36   35   37   36   38   37   39   38   40   39
N(*)      =   41   40   42   41   43   42   44   43   45   44   46   45   47   46   48   47   49   48   49
A(*,LL)   = 0.53 0.0  1.05 0.53 1.58 1.05 2.11 1.58 2.63 2.11 3.16 2.63 3.68 3.16 4.21 3.68 4.74 4.21 5.26 4.74
A(*,LL)   = 5.79 5.26 6.32 5.79 6.84 6.32 7.37 6.84 7.89 7.37 8.42 7.89 8.95 8.42 9.47 8.9510.00 9.4710.5310.00
A(*,LL)   =11.0510.5311.5811.0512.1111.5812.6312.1113.1612.6313.6813.1614.2113.6814.7414.2115.2614.7415.7915.26
A(*,LL)   =16.3215.7916.8416.3217.3716.8417.8917.3718.4217.8918.9518.4219.4718.9520.0019.4720.5320.0021.0520.53
A(*,LL)   =21.5821.0522.1121.5822.6322.1123.1622.6323.6823.1624.2123.6824.7424.2125.2624.7425.7925.2625.79
A(*,LL)   = 0.53 1.11 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
A(*,LL)   = 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
A(*,LL)   = 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
A(*,LL)   = 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
A(*,LL)   = 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53 0.53
A(*,LL)   = 0.53 0.0  0.0  0.53 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.53 0.0  0.0  0.53 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
A(*,LL)   = 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

Fig. 4 Sample Output from CHAIN Subroutine

```
RESULTS OF ROTATION SAMPLING FOR A MARKOV CHAIN

  NO. OF STATES                          =        50
  INITIAL STATE                          =         0
  ABSORBING STATE                        =         0
  TOTAL NO. OF (I,J) PAIRS               =        99
  NO. OF DESCRIPTORS                     =         4
  NO. OF CORRELATED MICROREPLICATIONS    =    131072
  NO. OF INDEPENDENT MACROREPLICATIONS   =         8
  INITIAL SEED                           =   1234567
  FINAL SEED                             = 590573803
  BLOCKING FACTOR                        =     40000

SAMPLE MEAN VECTOR
******************

        1              2              3              4              5              6              7              8              9

  0.96593614D 02 0.11052404D 02 0.99996355D 00 0.89989158D 00

SAMPLE COEFFICIENTS OF VARIATION
********************************

        1              2              3              4              5              6              7              8              9

  0.25163951D-02 0.70228364D-03 0.59198837D-04 0.13659310D-03

SAMPLE COVARIANCE/CORRELATION MATRIX
************************************

        1              2              3              4              5              6              7              8              9

1 0.59081904D-01 0.17939022D-02 0.37090189D-05 0.57659548D-05
2 0.95082860D 00 0.60247435D-04 0.19277545D-06 0.34303695D-06
3 0.25777124D 00 0.41955133D 00 0.35024468D-08 0.70713160D-08
4 0.19298566D 00 0.35954452D 00 0.97181581D 00 0.15109075D-07
```

Fig. 4 (*Continued*)

```
FIRST PASSAGE TIME (T) DISTRIBUTION
***********************************

SAMPLE MEAN(T) = 0.19886985D 02     SAMPLE VARIANCE(T) = 0.64011433D 04

                        MASS FUNCTION                                            DISTRIBUTION FUNCTION

                            VARIANCE        COEFFICIENT                            VARIANCE        COEFFICIENT
  I      PR(T=I)            OF PR(T=I)      OF VARIATION       PR(T<=I)            OF PR(T<=I)     OF VARIATION
 ---     -------            ----------      ------------       --------            -----------     ------------
  1      0.0                0.0             0.0                0.0                 0.0             0.0
  2      0.52631664D 00     0.19489172D-11  0.26524640E-05     0.52631664D 00      0.19489172D-11  0.26524640E-05
  3      0.0                0.0             0.0                0.52631664D 00      0.19489172D-11  0.26524640E-05
  4      0.13121223D 00     0.15591338D-11  0.95162832E-05     0.65752888D 00      0.40277623D-11  0.30522287E-05
  5      0.0                0.0             0.0                0.65752888D 00      0.40277623D-11  0.30522287E-05
  6      0.65426826D-01     0.19489172D-11  0.21337357E-04     0.72295570D 00      0.15591338D-11  0.17271495E-05
  7      0.0                0.0             0.0                0.72295570D 00      0.15591338D-11  0.17271495E-05
  8      0.40778160D-01     0.29883397D-11  0.42392334E-04     0.76373386D 00      0.50671848D-11  0.29474149E-05
  9      0.0                0.0             0.0                0.76373386D 00      0.50671848D-11  0.29474149E-05
 10      0.28466225D-01     0.50671848D-11  0.79077596E-04     0.79220009D 00      0.36379788D-11  0.24076598E-05
 11      0.0                0.0             0.0                0.79220009D 00      0.36379788D-11  0.24076598E-05
 12      0.21290779D-01     0.19489172D-11  0.65569984E-04     0.81349087D 00      0.29883397D-11  0.21250162E-05
 13      0.0                0.0             0.0                0.81349087D 00      0.29883397D-11  0.21250162E-05
 14      0.16678810D-01     0.29883397D-11  0.10364536E-03     0.83016968D 00      0.62365351D-11  0.30081865E-05
 15      0.0                0.0             0.0                0.83016968D 00      0.62365351D-11  0.30081865E-05
 16      0.13518333D-01     0.29883397D-11  0.12787682E-03     0.84368801D 00      0.29883397D-11  0.20489579E-05
 17      0.0                0.0             0.0                0.84368801D 00      0.29883397D-11  0.20489579E-05
 18      0.11230469D-01     0.20788450D-11  0.12838467E-03     0.85491848D 00      0.50671848D-11  0.26330472E-05
 19      0.0                0.0             0.0                0.85491848D 00      0.50671848D-11  0.26330472E-05
 20      0.95224380D-02     0.90949470D-12  0.10015022E-03     0.86444092D 00      0.83153801D-11  0.33358438E-05
 21      0.0                0.0             0.0                0.86444092D 00      0.83153801D-11  0.33358438E-05
 22      0.81996918D-02     0.15591338D-11  0.15228045E-03     0.87264061D 00      0.77956689D-11  0.31995669E-05
 23      0.0                0.0             0.0                0.87264061D 00      0.77956689D-11  0.31995669E-05
 24      0.71525574D-02     0.20788450D-11  0.20158105E-03     0.87979317D 00      0.77956689D-11  0.31735553E-05
 25      0.0                0.0             0.0                0.87979317D 00      0.77956689D-11  0.31735553E-05
 26      0.63123703D-02     0.40277623D-11  0.31793560E-03     0.88610554D 00      0.18579677D-10  0.48644479E-05
 27      0.0                0.0             0.0                0.88610554D 00      0.18579677D-10  0.48644479E-05
 28      0.56200027D-02     0.19489172D-11  0.24840469E-03     0.89172554D 00      0.18189894D-10  0.47828180E-05
 29      0.0                0.0             0.0                0.89172554D 00      0.18189894D-10  0.47828180E-05
 30      0.50439835D-02     0.90949470D-12  0.18907165E-03     0.89676952D 00      0.16500832D-10  0.45297274E-05
 31      0.0                0.0             0.0                0.89676952D 00      0.16500832D-10  0.45297274E-05
 32      0.45585632D-02     0.20788450D-11  0.31628809E-03     0.90132809D 00      0.13382565D-10  0.40586983E-05
 33      0.0                0.0             0.0                0.90132809D 00      0.13382565D-10  0.40586983E-05
 34      0.41465759D-02     0.20788450D-11  0.34771324E-03     0.90547466D 00      0.22737368D-10  0.52661562E-05
 35      0.0                0.0             0.0                0.90547466D 00      0.22737368D-10  0.52661562E-05
 36      0.37879944D-02     0.20788450D-11  0.38062874E-03     0.90926266D 00      0.17540255D-10  0.46060486E-05
 37      0.0                0.0             0.0                0.90926266D 00      0.17540255D-10  0.46060486E-05
 38      0.34790039D-02     0.20788450D-11  0.41443459E-03     0.91274166D 00      0.17540255D-10  0.45884926E-05
 39      0.0                0.0             0.0                0.91274166D 00      0.17540255D-10  0.45884926E-05
 40      0.87258339D-01     0.17540255D-10  0.47996655E-04     0.10000000D 01      0.0             0.0
```

Fig. 4. (*Continued*)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER UNC/ORSA/TR-82/3	2. GOVT ACCESSION NO. AD-A115451	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Simulating a Markov Chain with a Super-efficient Sampling Method		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George S. Fishman		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0302
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of North Carolina Chapel Hill, North Carolina 27514		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research Program Office of Naval Research Arlington, VA		12. REPORT DATE April, 1982
		13. NUMBER OF PAGES 39
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Markov chain Monte Carlo Simulation Variance reduction		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper describes an algorithm and a FORTRAN subprogram, CHAIN, for simulating the behavior of an (n+1) state Markov chain using a variance reducing technique called *rotation sampling*. The simulation of k *microreplications* is carried out in parallel at a mean cost $\le O(\ln k)$ and with variances of sample quantities of interest $\le O((\ln k)^2/k^2)$. The program allows for independent *macroreplications*, each of k microreplications, in order to facilitate estimation of the variances of sample quantities of interest. The paper describes theoretical results that underlie the algorithm and program in Section 1 and		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. presents applications of interest for first passage time and steady-state distributions in Section 2. Section 3 describes the algorithm and CHAIN and an example in Section 4 illustrates how CHAIN works in practice. Second 5 describes the options available for restarting the simulation.

DAT
ILM